

Here's the functional hierarchy for each role, formatted as it would appear in a Scope of Work document:

Functional Hierarchy of Participant Role:

1. Account Management:

- **Sign Up:**
 - **User Information Collection:** Implement a form to collect user details (name, email, password, location, etc.) with validation rules for data integrity.
 - **Team Selection:** Provide a dropdown menu or selection interface for users to choose a team or opt for individual participation.
 - **Terms and Conditions Display:** Display the terms and conditions document with a checkbox for user agreement.
 - **Privacy Policy Display:** Display the privacy policy document with a checkbox for user agreement.
- **Login:**
 - **Email/Password Authentication:** Implement secure authentication using bcrypt or a similar hashing algorithm for password storage.
 - **Forgot Password Functionality:** Implement a password reset feature that sends a reset link to the user's email address.
 - **Social Login Integration:** (Optional) Integrate with popular social media platforms (e.g., Facebook, Google) for user login.
- **Profile Management:**
 - **Edit Profile:** Implement a form for users to update their personal information with validation rules.
 - **Change Password:** Implement a password change feature with confirmation and security measures.
 - **Notification Settings:** Provide a settings menu for users to customize notification preferences (e.g., email, push notifications).
- **Logout:** Implement a logout functionality that clears user session data and redirects to the login page.

2. Trip Tracking:

- **Trip Logging:**
 - **Manual Entry:** Implement a form for users to manually input trip details (date, distance, mode of transport, trip type) with validation rules.
 - **GPS Tracking:** Integrate with a GPS API (e.g., Google Maps API) to enable automatic trip tracking.
 - **Trip Type Selection:** Provide a dropdown menu or selection interface for users to choose the trip type (work, errands, social, other).
 - **Mode of Transport Selection:** Provide a dropdown menu or selection interface for users to choose the mode of transport (walk, bike, run, e-bike, skate, other).
 - **Start/End Trip Functionality:** Implement buttons or controls to initiate and end trip tracking.
- **Trip History:**
 - **View Past Trips:** Display a list of logged trips with details (date, distance, mode, type) in a user-friendly format.
 - **Edit/Delete Trips:** Implement functionality for users to modify or remove logged trips with appropriate authorization checks.
 - **Filter Trips:** Implement filtering options for users to sort and filter trips by date, type, mode, etc.

3. Data & Insights:

- **Personal Dashboard:**
 - **Cumulative Data Display:** Display total trips, mileage, gas saved, CO2e reduced, and money saved for the current season in a visually appealing format.
 - **Calendar Year Data Display:** Implement functionality to switch to view cumulative data for the current calendar year.
 - **Lifetime Data Display:** Implement functionality to switch to view cumulative data since account creation.
- **Leaderboards:**
 - **Individual Leaderboard:** Display a ranked list of participants based on cumulative data for the current season with user information (name, team, etc.).
 - **Team Leaderboard:** Display a ranked list of teams based on cumulative data for the current season with team information (name, members, etc.).
 - **Historical Leaderboards:** Implement functionality to access leaderboards from previous seasons.

4. Social Features:

- **Team Chat:** Implement a chat feature within teams using a messaging API (e.g., Firebase, Twilio) with message history and notification capabilities.
- **Social Sharing:** Integrate with social media APIs (e.g., Facebook, Twitter) to allow users to share their progress and achievements.
- **Challenges:** Implement a challenge system with various challenge types, participation mechanisms, and reward systems.

5. Rewards & Incentives:

- **Points System:** Implement a points system that awards points based on trip logging and participation with clear point allocation rules.
- **Rewards Redemption:** Implement a system for users to redeem points for prizes or discounts with a catalog of available rewards.
- **Notifications:** Implement a notification system to inform users about new challenges, rewards, and program updates using push notifications or email.

Functional Hierarchy of Team Member Role:

1. Team Management:

- **Join Team:** Implement a feature for users to join an existing team with a search function and team details display.
- **Create Team:** Implement a feature for users to create a new team with a form for team name, description, and member selection.
- **Leave Team:** Implement a feature for users to exit a team with confirmation and data update.

2. Team Communication:

- **Team Chat:** Implement a chat feature within teams using a messaging API (e.g., Firebase, Twilio) with message history and notification capabilities.
- **Team Leaderboard:** Display the team's ranking and individual member contributions in a user-friendly format.

3. Team Challenges:

- **Participate in Team Challenges:** Implement a system for teams to participate in challenges with specific rules and rewards.
- **Contribute to Team Goals:** Implement a system for team members to contribute to the team's overall progress.

Functional Hierarchy of Admin/Super User Role:

1. User Management:

- **Approve/Deny Teams:** Implement a system for admins to review and approve or deny team creation requests with a notification system for applicants.
- **Manage User Accounts:** Implement a user management interface for admins to create, edit, and delete user accounts with appropriate authorization checks.
- **Assign/Remove Team Members:** Implement functionality for admins to add or remove users from teams with data update and notification capabilities.
- **Reset Passwords:** Implement a password reset feature for admins to reset user passwords with security measures.
- **Export User Data:** Implement a data export feature for admins to download user data in a specified format (e.g., CSV, JSON).

2. Trip Management:

- **Edit Trips:** Implement a feature for admins to modify trip details for individual participants with appropriate authorization checks.
- **Delete Trips:** Implement a feature for admins to remove trips from the system with confirmation and data update.

3. Program Management:

- **Manage Challenges:** Implement a challenge management interface for admins to create, edit, and manage challenges with a system for setting rules, rewards, and deadlines.
- **Set Rewards:** Implement a system for admins to define rewards for challenges and achievements with a catalog of available rewards.
- **Manage Notifications:** Implement a notification system for admins to send notifications to users using push notifications or email.
- **Reset Dashboard:** Implement a feature for admins to reset the dashboard data at the start of each season with data update and confirmation.

4. Data & Analytics:

- **Access Backend Database:** Provide admins with secure access to the app's database for data analysis and reporting.
- **Generate Reports:** Implement a reporting system for admins to generate reports on user activity, program performance, and other relevant metrics in various formats (e.g., charts, tables).

Functional Hierarchy of App Backend:

1. User Authentication:

- **User Registration:** Implement a secure registration process with data validation and password hashing.
- **User Login:** Implement a secure login process with email/password authentication and session management.
- **Password Management:** Implement password reset functionality and security measures for password changes.

2. Data Storage:

- **Database Design:** Design and implement a database schema to store user data, trip logs, team information, challenge data, and other relevant information.
- **Data Security:** Implement security measures to protect sensitive data from unauthorized access, modification, or deletion.
- **Data Backup and Recovery:** Implement a backup and recovery plan to ensure data integrity and availability.

3. Data Processing:

- **Trip Data Processing:** Process trip data to calculate metrics (mileage, gas saved, CO2e reduced, money saved) and update user dashboards.
- **Leaderboard Calculation:** Calculate and update leaderboards based on user data and challenge rules.
- **Data Aggregation and Reporting:** Aggregate data for reporting purposes and generate reports on user activity, program performance, and other relevant metrics.

4. API Management:

- **API Design:** Design and implement APIs for interacting with the app's functionalities (e.g., user authentication, trip logging, data retrieval).
- **API Documentation:** Provide clear and comprehensive documentation for the APIs to facilitate integration with the mobile app.
- **API Security:** Implement security measures to protect APIs from unauthorized access and malicious attacks.

Functional Hierarchy of Push Notification Service:

1. Notification Scheduling:

- **Notification Triggers:** Define triggers for sending notifications (e.g., new challenges, reward updates, trip reminders).
- **Notification Scheduling:** Implement a system for scheduling notifications based on time, events, or user preferences.

2. Notification Delivery:

- **Push Notification Integration:** Integrate with a push notification service (e.g., Firebase Cloud Messaging, OneSignal) to send notifications to users.
- **Notification Content:** Implement a system for creating and customizing notification content with relevant information and actions.

3. Notification Tracking:

- **Delivery Tracking:** Track the delivery status of notifications to ensure successful delivery.
- **Engagement Tracking:** Track user engagement with notifications (e.g., open rates, click-through rates).

Functional Hierarchy of Mapping and Geolocation Service:

1. Route Planning:

- **Geocoding:** Implement geocoding functionality to convert addresses or place names into geographic coordinates.
- **Route Calculation:** Integrate with a mapping API (e.g., Google Maps API) to calculate routes based on user location and destination.
- **Route Optimization:** Implement algorithms to optimize routes based on factors like distance, traffic, and user preferences.

2. Navigation Assistance:

- **Turn-by-Turn Navigation:** Implement turn-by-turn navigation guidance using a mapping API with voice prompts and visual cues.
- **Real-Time Traffic Updates:** Integrate with real-time traffic data to provide users with updated traffic conditions and alternative routes.

3. Location Tracking:

- **GPS Tracking:** Implement GPS tracking to track user location for trip logging and other purposes.
- **Location Accuracy:** Implement mechanisms to improve location accuracy and handle GPS signal issues.
- **Location Permissions:** Implement a system for requesting and managing user location permissions.

This detailed functional hierarchy, formatted for a Scope of Work document, provides a comprehensive technical breakdown of each role's responsibilities and functionalities. It ensures that the development team has a clear understanding of the project requirements and can effectively implement the Sole Power mobile app.